INVITED PAPER

Multirobot Simultaneous Localization and Mapping Using Manifold Representations

Data from exploring robots can be used to map individual robot paths separately; where robots meet, the paths may be merged to form a larger map.

By Andrew Howard, *Member IEEE*, Gaurav S. Sukhatme, *Senior Member IEEE*, and Maja J. Matarić, *Senior Member IEEE*

ABSTRACT | This paper describes a novel representation for two-dimensional maps, and shows how this representation may be applied to the problem of multirobot simultaneous localization and mapping. We are inspired by the notion of a manifold, which takes maps out of the two-dimensional plane and onto a surface embedded in a higher-dimensional space. The key advantage of the manifold representation is self-consistency: when closing loops, manifold maps do not suffer from the "cross over" problem exhibited in planar maps. This self-consistency, in turn, facilitates a number of important capabilities, including autonomous exploration, search, and retro-traverse. It also supports a very robust form of loop closure, in which pairs of robots act collectively to confirm or reject possible correspondence points. In this paper, we develop the basic formalism of the manifold representation, show how this may be applied to the multirobot simultaneous localization and mapping problem, and present experimental results obtained from teams of up to four robots in environments ranging in size from 400 to 900 m^2.

KEYWORDS | Exploration and search; multirobot systems; simultaneous localization and mapping (SLAM)

Manuscript received March 1, 2005; revised March 1, 2006. This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant 4400057784 SDR (Software for Distributed Robotics; in conjunction with SAIC, Telcordia, and the University of Tennessee, Knoxville) and Grant 5-39509-A (Mobile Autonomous Robot Software; in conjunction with the University of Pennsylvania, Georgia Institute of Technology, and BBN Technologies). This work was conducted at the Computer Science Department at the University of Southern California, Los Angeles, CA 90089-0781 USA.

A. Howard is with the Jet Propulsion Laboratory of the California Institute of Technology, 4800 Oak Grove Drive Pasadena, CA 91109 USA (e-mail: abhoward@robotics.jpl.nasa.gov).

G. S. Sukhatme and **M. J. Matarić** are with the Computer Science Department at the University of Southern California, Los Angeles, CA 90089-0781 USA (e-mail: gaurav@cs.usc.edu; mataric@cs.usc.edu).

Digital Object Identifier: 10.1109/JPROC.2006.876922

I. INTRODUCTION

Many mobile robot tasks, such as exploration and search, are performed in environments that are only partially mapped. As a result, these autonomous behaviors must be performed concurrently with a simultaneous localization and mapping process (SLAM). In this paper, we develop a map representation that remains *self-consistent* with respect to common navigation problems (such as path planning), while also supporting the simultaneous localization and mapping process. Standard planar maps are ill-suited for this purpose, due their tendency to become confused (at least temporarily) in environments containing loops. Consider, for example, the situation shown in Fig. 1: as the robot traverses a partial loop, the path of the robot crosses over itself. While this inconsistency may be ultimately be resolved by moving further along the loop, the planar map cannot be used for path-planning in the interim. In contrast, under the same conditions, a *manifold* representation remains entirely self-consistent. Robots can construct paths, for example, as long as those paths remain embedded in the manifold. Furthermore, if the robot does ultimately close the loop, the manifold can be collapsed to recover a self-consistent planar representation.

The manifold representation facilitates a number of interesting capabilities. For example, using incremental mapping alone (i.e., no loop closures), a robot can always retro-traverse to any previously visited location (or, more precisely, to any point on the manifold). In this case, the same location in the world may be represented more than once in the manifold: if the robot traverses a loop in one direction, for example, the manifold will develop a spiral structure, with the same locations being repeated over and over again. In spite of this ambiguity, however, the robot can always retro-traverse by traveling back along the spiral structure.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAR 2006**	2. REPORT TYPE	3. DATES COVERED
4. TITLE AND SUBTITLE **Multirobot Simultaneous Localization and Mapping Using Manifold Representations**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Southern California ,Department of Computer Science,Los Angeles,CA,90089-0781**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper describes a novel representation for two-dimensional maps, and shows how this representation may be applied to the problem of multirobot simultaneous localization and mapping. We are inspired by the notion of a manifold, which takes maps out of the two-dimensional plane and onto a surface embedded in a higher-dimensional space. The key advantage of the manifold representation is selfconsistency when closing loops, manifold maps do not suffer from the Bcross over[problem exhibited in planar maps. This self-consistency, in turn, facilitates a number of important capabilities, including autonomous exploration, search, and retro-traverse. It also supports a very robust form of loop closure, in which pairs of robots act collectively to confirm or reject possible correspondence points. In this paper, we develop the basic formalism of the manifold representation, show how this may be applied to the multirobot simultaneous localization and mapping problem, and present experimental results obtained from teams of up to four robots in environments ranging in size from 400 to 900 m2.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**10**	

The many-to-one relationship between points on the manifold and points in the world gives rise to a second interesting capability: *lazy loop closure*. Loop closure is the most difficult part of the simultaneous localization and mapping process: in order to close a loop, one must decide that two points in the map correspond to the same point in the world (this is the data-association problem). In the manifold representation, such decisions can be indefinitely delayed, without risking map consistency; thus, one may wait until robots acquire more information to conclusively establish the correspondence (or lack thereof) between two points. In the multirobot context, this allows us to take *active* steps to discover correspondence points, using pairs of robots acting in concert. Thus, for example, a pair of robots can arrange a rendezvous at two points on the manifold that may or may not represent the same location in the world: if the robots meet, the points match and the loop is closed; if they fail to meet, the points are distinct and there is no loop.

This paper makes no attempt to cover all aspects the manifold representation outlined above. Instead, we restrict ourselves to introducing the basic methodology and applying it to the specific problem of multirobot mapping. We take maximum likelihood estimation (MLE) techniques that have previously been applied to simultaneous localization and mapping [1], [2], and adapt those techniques for the manifold representation. For validation, we present experimental mapping results from a number environments, using teams of up to four robots, under both manual and autonomous control.

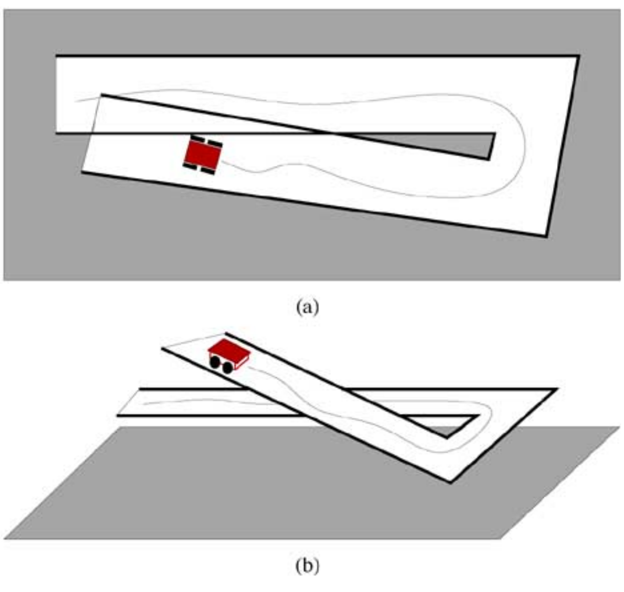


Fig. 1. ***Illustration of a partially closed loop. (a) Planar representation. (b) Manifold representation.***

II. RELATED WORK

Thrun's survey paper [3] identifies a relatively small set of probabilistic methods underlying most recent SLAM algorithms: these include Kalman filters [4], [5], expectation maximization [6], incremental maximum likelihood [2], and various hybrid methods [7]. Recent work on FastSLAM algorithms (which approximate the full posterior distribution over maps using a particle filter) should be added to this list [8], [9]. The approach described in this paper makes use of both incremental MLE (for fine-scale localization) and Lu-and-Milios-style global map alignment [1]. We also make use of local map patches to enforce global/topological map consistency; this is similar in concept, if not in detail, to the approach taken in the *Atlas* framework [10], and in hybrid extensions to the Spatial Semantic Hierarchy [11].

Existing SLAM methods have known limitations, the most important of which is sensitivity to false data associations. The manifold representation is designed, in part, to overcome this limitation: it provides a mechanism whereby data-association decisions can be postponed until such time as they can be made with high confidence. Hähnel [12] describes an alternative—and possibly more powerful—approach to this problem, in which bad data-association decisions may be retracted by stepping back through a tree-structured representation of possible maps.

It should be noted that the research described in this paper was spurred by a very specific programmatic challenge: to deploy a large number of robots into an unexplored building, map the building interior, detect and track intruders, and transmit all of the above information to a remote operator. The multirobot systems built to meet this challenge are described in [13] and [14].

III. MAPPING ON A MANIFOLD: CORE CONCEPTS

The key conceptual difficulty with manifold mapping lies in the representation of the manifold itself. In principle, the manifold is an arbitrarily complex structure with varying local curvature; in practice, the representation must be discrete, and hence some degree of approximation and linearization is inevitable. In this section, we develop the basic concepts, definitions, and notation used in our approximated representation.

A. Patches

The manifold is discretized by dividing it into a set of overlapping *patches*, each of which has finite extent and defines a local (planar) coordinate system. Let Π denote the set of such patches; we make the following definition:

$$\Pi = \{\pi\} : \pi = (\theta, s) \tag{1}$$

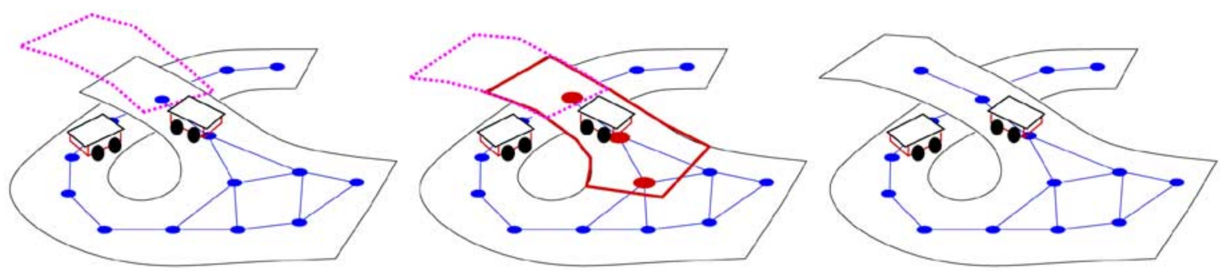


Fig. 2. *Incremental localization and mapping. (a) and (b) A new laser scan (dotted polygon) is fitted against the robot's local map (solid polygon), to generate a corrected robot pose estimate. (c) If the laser scan covers unexplored regions of the manifold, a new patch is added to the map.*

where an individual patch π consists of a *free-space polygon* s describing the extent of the patch,[1] and a *projected planar pose* θ that defines the patch-local coordinate system (θ is obtained by projecting the origin of the patch onto a canonical plane).

Given these definitions, the pose of any object on the manifold can subsequently be described by a tuple $\rho = (\pi, r)$ specifying a particular patch π and the pose r of the object with respect to that patch (r must lie inside the patch polygon s). Importantly, since patches may overlap, the tuple ρ need not be unique: one can also write down the same object's pose as $\bar{\rho} = (\bar{\pi}, \bar{r})$ where $\bar{r}$ is the pose relative to some overlapping patch $\bar{\pi}$. Two questions naturally arise from this apparent ambiguity: how does one establish that the tuples ρ and $\bar{\rho}$ represent the same pose on the manifold, and how does one transform a pose specified with respect to patch π into a pose specified with respect an overlapping patch $\bar{\pi}$? Consider the manifold illustrated in Fig. 2: each point on the manifold will project onto exactly one point on an imaginary horizontal plane, and, conversely, some points on the plane will project onto multiple points on the manifold. This observation leads to the following condition: the tuples ρ and $\bar{\rho}$ represent the same point on the manifold if and only if the projections of the polygons s and $\bar{s}$ overlap, and the projections of r and $\bar{r}$ are identical. Mathematically, this condition is stated as follows:

$$(s \oplus \theta) \cap (\bar{s} \oplus \bar{\theta}) \neq \emptyset \text{ and } r \oplus \theta - \bar{r} \oplus \bar{\theta} = 0 \qquad (2)$$

where $\oplus$ is a *coordinate transform operator*. Given a projected pose θ and patch-relative pose r, the expression $r \oplus \theta$ yields the corresponding projected pose q. One can also define the inverse operator $\ominus$: given two projected poses θ and q, the inverse expression $q \ominus \theta$ recovers the patch-relative pose r. These operators obey the normal rules for algebraic associativity, but do not commute.

From the identity expressed in (2), one can trivially derive the coordinate transform equations for overlapping patches

$$\bar{r} = r \oplus \theta \ominus \bar{\theta} \text{ and } r = \bar{r} \oplus \bar{\theta} \ominus \theta. \qquad (3)$$

Collectively, (2) and (3) provide the necessary tools for working with manifold poses and their planar projections. Importantly, one can use these equations to construct paths on the manifold.

B. Relations

For concurrent localization and mapping, the projected poses of the patches Π are not known *a priori*; instead we have a set of *relations* that constrain the patches' *relative* pose (a scan-matching algorithm, for example, may establish point-to-point correspondences). Let Φ denote the set of pairwise relations between patches; we write

$$\Phi = \{\phi\} : \phi = (\pi, \bar{\pi}; x, \bar{x}, \sigma) \qquad (4)$$

where the relation ϕ implies that point x on patch π corresponds to point $\bar{x}$ on patch $\bar{\pi}$; σ is the uncertainty associated with the correspondence. One can write down similar definitions for point-to-line, line-to-line, relative range, and relative bearing relations.

[1]Strictly speaking, we use *polysolids* rather than polygons for representing free space, since polysolids form a group under the operations of union and intersection (polysolids can have holes). The term "polysolid" appears to have been coined by H. Maynard and L. Tavernini at the University of Texas at San Antonio; their work was never published, but is similar in concept to the *polygon sets* described in [15].

C. Fitting Patches

Given the above definitions, one can apply MLE techniques to find the set of projected poses $\Theta = \{\theta\}$ that is *most likely* to generate the observed set of relations $\Phi = \{\phi\}$. That is, MLE searches for the estimate $\hat{\Theta}$ that maximizes the conditional probability $P(\Phi|\Theta)$

$$\begin{aligned}\hat{\Theta} &= \arg\max_{\Theta} P(\Phi|\Theta)\\ &= \arg\max_{\Theta} \prod_{\phi\in\Phi} P(\phi|\Theta)\end{aligned} \tag{5}$$

where we make the additional assumption that the relations in Φ represent statistically independent observations. Applying the standard log-likelihood transformation to these equations, one can equivalently search for the Θ that *minimizes* the (negative) conditional log-likelihood $L(\Phi|\Theta)$

$$\hat{\Theta} = \arg\min_{\Theta} \sum_{\phi\in\Phi} L(\phi|\Theta). \tag{6}$$

This latter form is more convenient for most practical purposes. For point-to-point relations with Gaussian uncertainty, the log-likelihood for a single relation ϕ is given by

$$L(\phi|\Theta) = \frac{1}{2\sigma^2}(x \oplus \theta - \bar{x} \oplus \bar{\theta})^2 \tag{7}$$

where $x \oplus \theta$ denotes the projected pose of a point on patch π and $\bar{x} \oplus \bar{\theta}$ denotes the projected pose of the corresponding point on patch $\bar{\pi}$. Intuitively, one can visualize the two points as being pulled together by a simple spring.

In principle, the maximum likelihood estimate $\hat{\Theta}$ can be found by solving

$$0 = \sum_{\phi\in\Phi} \nabla L(\phi|\Theta). \tag{8}$$

In the case of two-dimensional mapping, however, the gradient terms on the right-hand side of (8) generally contain transcendental components, and hence there exists no closed-form solution. Fortunately, a range of numeric optimization techniques can used to find the minimum, including simple gradient descent and its more refined brethren, such as the Levenburg–Marquardt and Fletcher–Reeves algorithms [16].

The *confidence* in the estimate $\hat{\Theta}$ can be determined by inspecting the local curvature of $\nabla L(\Phi|\Theta)$ around $\hat{\Theta}$. Following the standard practice in the MLE literature [17], we write down the *stochastic Fisher information matrix* as

$$J(\Phi|\hat{\Theta}) = \sum_{\phi\in\Phi} \nabla^2 L(\phi|\hat{\Theta}). \tag{9}$$

The confidence interval $\hat{\sigma}_i$ on any component i of $\hat{\Theta}$ is then given by

$$\hat{\sigma}_i = c\sqrt{\left(J^{-1}(\Phi|\hat{\Theta})\right)_{ii}} \tag{10}$$

where c is an appropriate critical value (e.g., 1.96 for a 95% confidence interval), and the *ii* subscript denotes a particular component of the inverted Fisher information matrix. In practice, we are usually less interested in the absolute fit confidence than we are in the relative confidence between a given pair of patches; i.e., how well is the pose of patch π determined with respect to patch $\bar{\pi}$? This quantity can be determined using a sightly modified version of the above procedure: we treat the components of $\hat{\Theta}$ that correspond to patch π as constants, and eliminate the corresponding rows and columns from the Fisher information matrix. Inserting this reduced matrix into (10), we obtain the component-wise confidence intervals for every patch, relative to the "fixed" patch π. For the remainder of this paper, we will use $J(\Phi|\hat{\Theta})$ to denote this reduced matrix.

IV. MULTIROBOT MAPPING

We turn now to the specific problem of multirobot mapping, using the mathematical tools described in the previous section. This problem can be broken into three subproblems: incremental localization and mapping, loop closure, and island merging.

Incremental localization and mapping is the basic mode of operation for the mapping algorithm: as each robot moves through the environment, odometry and laser data are used to update the robot's current pose estimate and, under certain circumstances, to make incremental additions to the map. The basic process is illustrated in Fig. 2 and described in detail in the next section. Note that robots extend the map at the *edges* of the manifold only; a robot that is retro-traversing to a previously visited location will not add to the map.

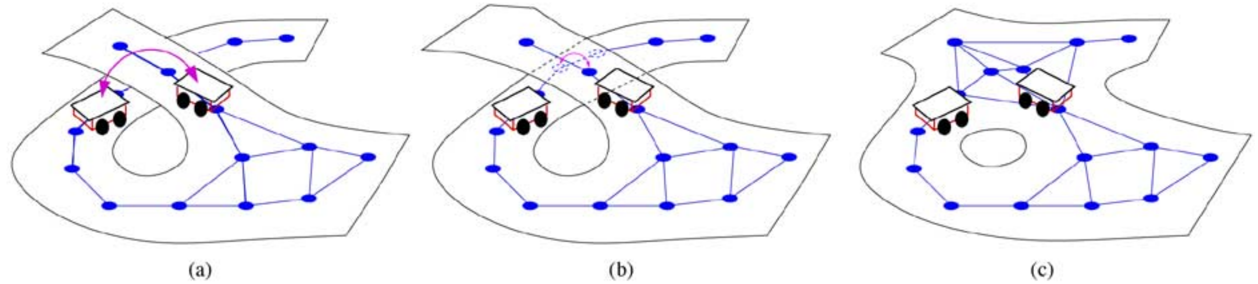


Fig. 3. *A loop closure triggered by a mutual observation. (a) & (b) Two robots observe each other, generating a new relation. (c) The change in topology is propagated through the manifold, inducing new relations and forcing patches to be refitted.*

This process is punctuated by two events that require *global* changes to the map: loop closure and island merging. Loop closure is the process whereby two widely separated regions of the map are brought together (see Fig. 3). In Section III-A, we showed that multiple points on the manifold may project onto the same point on the plane; in the context of mapping, this implies that two widely separated points on the manifold may in fact represent the same location in the world. Indeed, if one uses incremental mapping alone, any loops in the environment will be "unrolled" to form spiral structures, with the same series of locations repeating over and over again in the manifold. Loop closure, then, is the process whereby such repeated locations are identified, and the topology of the manifold is modified accordingly.

In a similar vein, island merging is the process whereby two unconnected regions of the manifold are combined into a single representation (see Fig. 4). In the context of multirobot mapping, there are two basic scenarios that give rise to such islands: robots enter the environment from separate locations, or robots enter the environment from the same location, but at different times. In either case, we proceed by building a separate island for each robot, and merging those islands only when a suitable correspondence point has been established.

The loop-closure and island-merging processes depend on our ability to uniquely identify a particular location in the world (the traditional data-association problem). In the case of single-robot mapping, there are two basic methods for making this identification: recognizing a unique feature associated with that location (including preplaced fiducials) or making plausible guesses based on patterns of nonunique features. These two methods have been well treated in the single-robot mapping literature [2], [6], [7], [12], and will not be covered here. Instead, we focus on a third method that is unique to the multirobot mapping domain: using the robots themselves as unambiguous landmarks. Whenever two robots sight one another—a process we refer to as *mutual observation*—we establish a correspondence between two points on the manifold; mutual observations can therefore be used to close loops and merge islands.

In the following sections, we describe the incremental mapping, loop-closure, and island-merging processes in detail. Note that, throughout this presentation, we assume

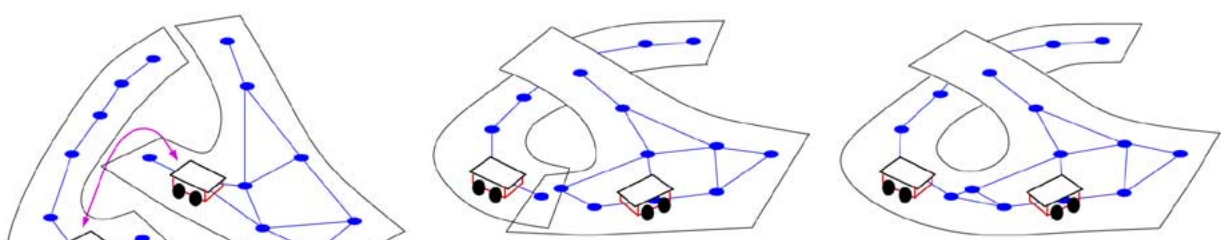


Fig. 4. *Merging islands after a mutual observation. (a) Two robots observe each other, generating a new relation. (b) & (c) The two islands are roughly aligned, and the change in topology is propagated through the manifold.*

that the mapping algorithm is *centralized*; i.e., data from all of the robots is communicated to a common location, where it is assembled to form a map.

A. Incremental Localization and Mapping

Incremental localization and mapping is performed independently and concurrently for each robot on the team. Two pieces of sensor data are used in this process: odometry data (which measures changes in the robot's pose) and laser scan data (which measures the range and bearing of nearby features).

Let $o_{tt'}$ be the measured (odometric) change in pose between times t and t', and let $s_{t'}$ be the laser scan that is subsequently recorded. If $\rho_t = (\pi_t, r_t)$ is the robot pose estimate at some time t, the updated robot pose estimate $\rho_{t'} = (\pi_{t'}, r_{t'})$ can be determined as follows.

1) Create a new patch $\pi^* = (\theta^*, s^*)$ such that

$$\theta^* \leftarrow o_{tt'} \oplus r_t \oplus \theta_t \text{ and } s^* \leftarrow s_{t'}, \tag{11}$$

i.e., the projected pose θ^* is computed by combining the measured (odometric) change in pose $o_{tt'}$ with the robot's current pose estimate ρ_t.

2) Create a *local map* around the current patch π_t; the local map is the set of patches Π^* that are both nearby (in the planar projection) and *well fitted* with respect to π_t; that is, Π^* contains all patches π that satisfy the condition

$$1.96\sqrt{\left(J^{-1}(\Phi|\hat{\Theta})\right)_{\theta\theta}} < \epsilon. \tag{12}$$

The notation here requires a little explanation: $J(\Phi|\hat{\Theta})$ is the Fisher information matrix computed for patch π_t (see Section III-C); the patch π is included in the local map only if the confidence interval on every component of θ is less than some threshold ϵ.

3) Match features in the new patch π^* against features in the local map Π^*. We omit the details of the scan-matching algorithm and assume only that it produces some set of relations Φ^*.

4) Use MLE to fit the new patch against the local map; i.e., find the projected pose θ^* that is most likely to give rise to the observed relations Φ^*

$$\theta^* \leftarrow \arg\min_{\theta} L(\Phi^*|\theta). \tag{13}$$

The minimum value is found using numeric optimization.

5) Compute the new robot pose estimate $\rho_{t'} = (\pi_{t'}, r_{t'})$ by projecting the new patch back into the manifold

$$\begin{aligned} \pi_{t'} &\leftarrow \arg\min_{\pi\in\Pi^\dagger} \|\theta^* \ominus \theta\| \\ r_{t'} &\leftarrow \theta^* \ominus \theta_{t'} \end{aligned} \tag{14}$$

where $\Pi^\dagger$ is a subset of Π^* containing only those patches whose scan polygons overlap with π_t; the potential ambiguity in the projection is resolved by selecting the nearest patch from this set.

Steps 3 and 4 of the algorithm may be applied iteratively (EM style) to improve the quality of the fit.

The key step in this algorithm lies in the creation of the local map. In effect, that part of the manifold that is well localized with respect to the robot is projected onto a plane; the robot is then localized by fitting its laser scan against this planar projection. In this context, the choice of the threshold ϵ becomes crucial, since this parameter implicitly controls the number of patches included in the local map: if ϵ is too small, there may not be enough patches to adequately constrain the robot pose; if ϵ is too large, the local map may contain gross inconsistencies that lead to widely inaccurate pose estimates.

Having localized the robot, we may need to extend the map. There are a number of conditions that can trigger this process: e.g., the new patch is far from any of the existing patches in the local map, or the new patch covers' a significant area of the manifold that is not covered by the current local map Π^*. If none of these conditions are true, the patch π^* is discarded; otherwise, the patch and its relations are appended to the map. In this case, the robot pose estimate $\rho_{t'} = (\pi_{t'}, r_{t'})$ must also reset such that the robot lies at the origin of the new patch; i.e.,

$$\pi_{t'} \leftarrow \pi^* \text{ and } r_{t'} \leftarrow 0. \tag{15}$$

B. Loop Closure

The loop-closure algorithm is triggered by events (such as mutual observation) that generate new relationships between previously unrelated patches. The key challenge for this algorithm lies not in the integration of this singular relation into the map; rather, it lies in the integration of any *additional* relations that may be induced as the change is propagated through the map. Consider, for example, a pair of robots traveling in opposite directions around a circular environment. If the robots should fail to observe each other on the first few passes (and thus fail to close the loop), the manifold will develop a double spiral structure (one spiral for each robot). The loop-closure algorithm must be such that a single subsequent mutual observation will the collapse the *entirety* of both spirals into a single loop.

Our method for achieving this collapse is as follows: given a new relation, the algorithm propagates changes outwards from the closure point, alternating between inducing new relations and refitting the map. The process terminates only when no new relations can be induced. Let ϕ_{ab} denote a new relation between previously unrelated patches π_a and π_b; the algorithm is as follows.

1) Add the new relation ϕ_{ab} to the map and refit

$$\Theta \leftarrow \arg\min_{\Theta'} \sum_{\phi \in \Phi} L(\phi|\Theta'). \tag{16}$$

2) Create a queue of patches; initialize the queue with the set of all patches that are related to π_a or π_b.
3) Pop the first patch off the queue; call this patch π^*. Induce the local map Π^* for this patch using the procedure described in Section IV-A.
4) Use scan-matching to fit the patch π^* to the local map Π^*. From the set of relations Φ^* found by the scan-matching algorithm, eliminate those that are already in the map; i.e., Φ^* should represent the set of *new* relations for patch π.
5) If Φ^* is nonempty:
   a) add the new relations Φ^* to the map and refit as per step 1;
   b) add all patches that are related to π^* to the queue.

The process continues until the queue is empty.

Compared to incremental localization and mapping, the loop-closure algorithm is relatively expensive: refitting the entire map is nontrivial, and may be performed more than once for any given loop closure. Fortunately, closure events are relatively rare (their frequency depends on the number of loops in the environment). In addition, the loop-closure algorithm is executed only *once* for each loop in the environment; subsequent traversals of a loop will incur no penalty.

C. Island Merging

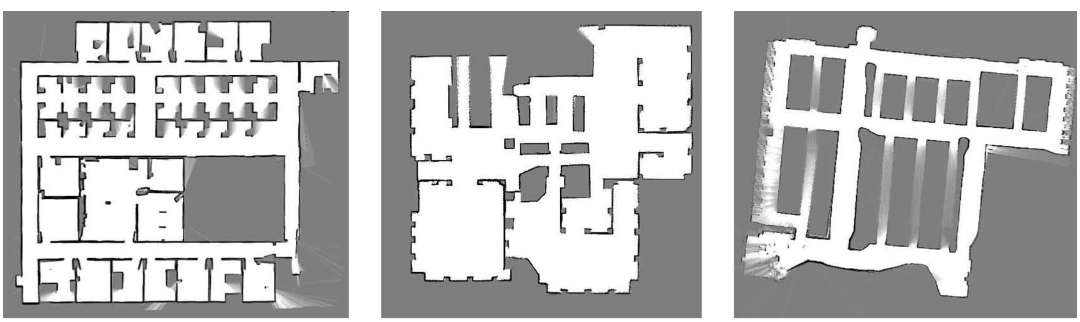
The island-merging algorithm is triggered by events that induce relations between patches belonging to separate islands (patches are said to belong to the same island only if and only if they are connected by some sequence of relations). In this case, as with loop closure, we must admit the possibility that there is substantial overlap between the two islands, and any changes made at the point where the islands are merged must be propagated throughout the map. The algorithm for island merging is therefore identical to that used for loop closure, with one exception: prior to merging, we treat the two islands as rigid bodies and quickly bring them into rough alignment, thus saving a great deal of time in the refitting process.

Compared with incremental mapping, island merging is a relatively expensive process. For a team of N robots, however, the algorithm will be executed at most $N-1$ times; moreover, since robots are likely to commence mapping from a relatively small number of initial locations, most of these mergers are trivial (i.e., involving islands with only a handful of patches).

V. EXPERIMENTS

The multirobot mapping approach described in this paper has been applied to a wide range of environments of varying size and complexity. Fig. 5 shows a selection of the final occupancy grid maps produced by the algorithm; these maps were generated autonomously, and in real time (the data sets used to generate these maps can be downloaded from the Radish [18] Web site).

Fig. 6 shows the result for one particularly challenging experiment conducted under external supervision (i.e., the environment and test conditions were selected by an



(a) (b) (c)

Fig. 5. ***Occupancy grids generated by the mapping package. (a) SAIC site A (two robots). (b) SAIC site B (four robots). (c) USC Science Library (four robots).***

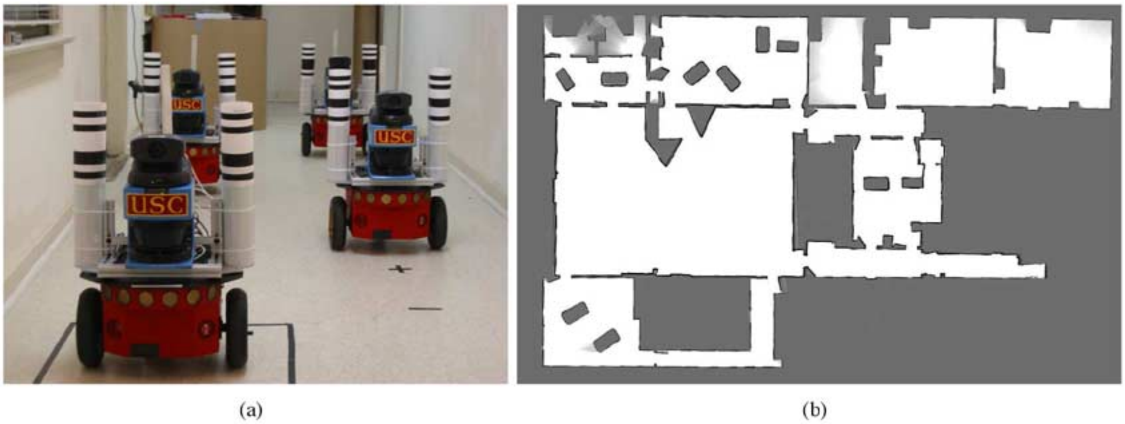


Fig. 6. ***(a) A four-robot mapping team; each robot carries a unique laser-visual bar code to facilitate mutual recognition. (b) Final occupancy grid produced during a multiple-robot, multiple-entry trial: two robots enter from the door at the right top, another two robots enter from the door at the right bottom (the relative pose of the two doorways is unknown). The environment is approximately 600 m^2 in area.***

independent group). Four robots were deployed into the test environment and executed an autonomous exploration and search algorithm. Importantly, the robots were deployed from *two different* entry points, and the relative pose of these points was unknown. Each pair of robots was therefore required to explore and map independently, until such time as the teams came within line of sight of one another. To facilitate mutual observation, each robot was equipped with a scanning laser-range finder, a camera, and a laser-visual bar code. Using the bar code, each robot can determine the identity, range and bearing of other robots; through the exchange of such observations, each *pair* of robots can determine their relative pose to within a few centimeters (over a distance of about 8 m).

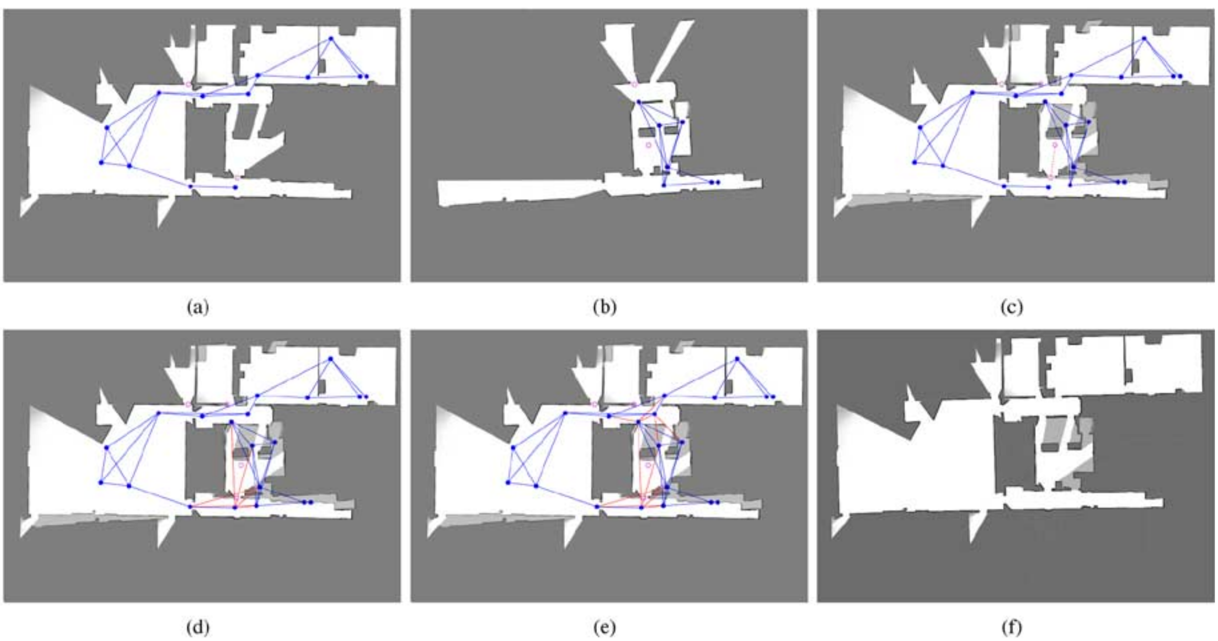


Fig. 7. ***(a) Occupancy grid generated by the pair of robots entering the environment from the top right. Open circles show the current robot locations; closed circles show the patch centers. (b) Occupancy grid generated by the pair of robots entering from the bottom left. Note that these maps have been reoriented to facilitate visual comparison; from the robot's perspective, the relative pose of the two maps is unknown. (c) A mutual observation (dotted line) allows the maps to be roughly aligned. (d) and (e) Relationships are incrementally propagated through the manifold. (f) Relaxed grid map using data from all four robots.***

Fig. 7 illustrates the evolution of the map over the course of the experiment. Initially, the two teams construct separate maps (or islands) as shown in Fig. 7(a) and (b). After a few minutes of autonomous exploration, however, one of the robots from the first team encounters a robot from the second (the two robots make a mutual observation and determine their relative pose). This triggers an island-merging event (Section IV-C), which first brings the two maps into rough alignment [Fig. 7(c)], then iteratively propagates new relationships between the maps [Fig. 7(d) and (e)]. The final combined map after island merging is shown in Fig. 7(f). Exploration continues with all four robots, ultimately producing the map shown in Fig. 6(b).

It should be noted that this map was generated online and in real time (i.e., the time required to produce the map was less than the time required for the robots to explore the environment). During island merging or loop closure, however, it is necessary to halt the robots for some tens of seconds to allow the mapping process to "catch up."

VI. CONCLUSION AND FURTHER WORK

While the manifold techniques described in this paper may be used to generate maps from robots under manual control, our key motivation in designing this representation is to support *autonomous* behaviors for incompletely mapped environments. As a result, our current research is heavily focused on this topic. To date, we have created an autonomous exploration algorithm that exploits the manifold map to direct and coordinate multiple robots; our near-term aim is to extend this algorithm to include *planned rendezvous* [19]: i.e., the use of robots in pairs to explore regions of the manifold that appear similar.

The algorithm, as presented, also has some key limitations: since it relies on centralized processing, it is sensitive to communication failures and scales poorly with increasing team size (the largest experiment conducted to date employed just four robots). Developing a truly distributed version of this algorithm thus remains an open challenge.

REFERENCES

[1] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.

[2] J. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Computational Intelligence in Robotics and Automation (CIRA)*, 2000, pp. 318–325.

[3] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer, and B. Nebel, Eds. San Francisco, CA: Morgan Kaufmann, 2002.

[4] S. Borthwick and H. Durrant-Whyte, "Simultaneous localisation and map building for autonomous guided vehicle," in *Proc. IEEE/RSJ/GI Int. Conf. Intelligent Robots and Systems*, 1994, vol. 2, pp. 761–768.

[5] J. J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. IEEE/RSJ Int. Workshop Intelligent Robots and Systems*, 1991, vol. 3, pp. 1442–1447.

[6] S. Thrun, D. Fox, and W. Burgard, "A probabilistic approach to concurrent mapping and localisation for mobile robots," *Mach. Learn.*, vol. 31, no. 5, pp. 29–55, 1998, joint issue with Autonomous Robots.

[7] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 335–363, 2001.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *AAAI Nat. Conf. Artificial Intelligence*, Edmonton, AB, Canada, 2002.

[9] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003, vol. 1, pp. 206–211.

[10] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *IEEE Int. Conf. Robotics and Automation*, Taipei, Taiwan, 2003, pp. 1899–1906.

[11] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *IEEE Int. Conf. Robotics and Automation (ICRA 2004)*, pp. 4845–4851.

[12] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun, "Towards lazy data association in SLAM," presented at the 10th Int. Symp. Robotics Research (ISRR'03), Siena, Italy.

[13] A. Howard, L. E. Parker, and G. S. Sukhatme, "Experiments with a large heterogeneous mobile robot team: exploration, mapping, deployment and detection," *Int. J. Robot. Res.*, vol. 5–6, no. 25, pp. 431–447, 2006.

[14] K. Konolige, D. Fox, C. Ortiz, A. Agno, M. Eriksen, B. Limketkai, J. Ko, B. Morisset, D. Schulz, B. Stewart, and R. Vincent, "Centibots: Very large scale distributed robotic teams," presented at the 9th Int. Symp. Experimental Robotics (ISER-04), Singapore, Jun. 2004.

[15] B. R. Vatti, "A generic solution to polygon clipping," *Commun. ACM*, vol. 35, no. 7, pp. 57–63, 1992.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[17] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis* ser. Electrical and Computer Engineering: Digital Signal Processing, Reading, MA: Addison-Wesley, 1991.

[18] A. Howard and N. Roy. (2003). *Radish: The Robotics Data Set Repository*. [Online]. Available: http://radish.sourceforge.net

[19] N. Roy, and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," in *Auton. Robots*, vol. 11, no. 2, pp. 117–136, 2001.

ABOUT THE AUTHORS

Andrew Howard (Member, IEEE) received the B.Sc. (Hons) degree in physics and the Ph.D. degree in engineering from the University of Melbourne, Australia. He is a Senior Member of Technical Staff with the Computer Vision Group at the Jet Propulsion Laboratory, and Adjunct Assistant Professor with the Computer Science Department at the University of Southern California. His research interest include state estimation, localization, mapping and multirobot systems. He is cocreator of the Player/Stage/Gazebo open source software suite and founder of the Radish data-set repository.

Gaurav S. Sukhatme (Senior Member, IEEE) is Associate Professor of Computer Science (joint appointment in Electrical Engineering—Systems) at the University of Southern California (USC). He received his undergraduate education at IIT Bombay in Computer Science and Engineering, and M.S. and Ph.D. degrees in Computer Science from USC. He is the codirector of the USC Robotics Research Laboratory and the director of the USC Robotic Embedded Systems Laboratory which he founded in 2000. His research interests are in multirobot systems, networked robots, and sensor/actuator networks. He has published more than 120 papers in these and related areas. Sukhatme has served as PI on numerous NSF, DARPA and NASA grants. He is a member of AAAI, ACM, a senior member of IEEE, and a recipient of the NSF CAREER award. He has served on many conference program committees; he was program cochair of the first Robotics: Science and Systems (RSS) conference and is the general cochair for RSS 2006. He is the program cochair for IEEE ICRA 2008. He is the Associate Editor of *Autonomous Robots,* and an Associate Editor of the IEEE Transactions on Robotics and Automation, both leading journals in robotics. He is on the editorial board of a leading magazine in ubiquitous computing, *IEEE Pervasive Computing*.

Maja J. Matarić (Senior Member, IEEE) is Professor of Computer Science and Neuroscience at the University of Southern California (USC), founding director of the USC Center for Robotics and Embedded Systems (cres.usc.edu), and codirector of the USC Robotics Research Lab (robotics.usc.edu). She received a PhD in Computer Science and Artificial Intelligence from MIT in 1994, MS in Computer Science from MIT in 1990, and BS in Computer Science from the University of Kansas in 1987. She is a recipient of the Okawa Foundation Award, NSF Career Award, the MIT TR100 Innovation Award, the IEEE Robotics and Automation Society Early Career Award, the USC Viterbi School of Engineering Service Award and Junior Research Award, USC Provost's Center for Interdisciplinary Research Fellowship, and is featured in the documentary movie "Me & Isaac Newton." She is an associate editor of three major journals (*Int. Journal of Humanoid Robotics, Int. Journal on Autonomous Agents and MultiAgent Systems*, and *Adaptive Behavior*) and has served as editor of the IEEE Transactions on Robotics and Automation and of the *Int. Journal of AI Research*. She has served on numerous conference program committees, most recently on the Steering Committee of the IEEE ICRA 2005 and is the General CoChair of IEEE ICRA 2008. Prof. Matarić has published extensively in several areas of robotics. Her research is aimed at developing control algorithms that will enable robots to provide personal assistance individually (in convalescence, rehabilitation, training, and education) and in teams (in habitat monitoring and emergency response).